Building an Expert System for Evaluation of Commercial Cloud Services

Zheng Li
School of CS
NICTA and ANU
Canberra, Australia
Zheng.Li@nicta.com.au

Liam O'Brien
School of CS
CECS, ANU
Canberra, Australia
liamob99@hotmail.com

Rainbow Cai
Division of Information
Information Services, ANU
Canberra, Australia
Rainbow.Cai@anu.edu.au

He Zhang
School of CSE
NICTA and UNSW
Sydney, Australia
He.Zhang@nicta.com.au

***Abstract*—Commercial Cloud services have been increasingly supplied to customers in industry. To facilitate customers' decision makings like cost-benefit analysis or Cloud provider selection, evaluation of those Cloud services are becoming more and more crucial. However, compared with evaluation of traditional computing systems, more challenges will inevitably appear when evaluating rapidly-changing and user-uncontrollable commercial Cloud services. This paper proposes an expert system for Cloud evaluation that addresses emerging evaluation challenges in the context of Cloud Computing. Based on the knowledge and data accumulated by exploring the existing evaluation work, this expert system has been conceptually validated to be able to give suggestions and guidelines for implementing new evaluation experiments. As such, users can conveniently obtain evaluation experiences by using this expert system, which is essentially able to make existing efforts in Cloud services evaluation reusable and sustainable.**

***Index Terms*—Expert System; Cloud Computing; Commercial Cloud Service; Cloud Services Evaluation; Evaluation Experiences**

I. Introduction

Since Cloud Computing has become increasingly accepted as one of the most promising computing paradigms in industry [5], providing Cloud services also becomes an emerging business. An increasing number of providers have started to supply commercial Cloud services with different terminologies, definitions, and goals [21]. As such, evaluation of those Cloud services would be crucial for many purposes ranging from cost-benefit analysis for Cloud Computing adoption to decision making for Cloud provider selection. However, evaluation of commercial Cloud services is different to and more challenging than that of other computing systems. There are three main reasons for this:

- In contrast with traditional computing systems, the Cloud is relatively chaos [25]. There is still a lack of standard definition of Cloud Computing, which inevitably leads to market hype and also skepticism and confusion [27]. As a result, it is hard to point out the range of Cloud Computing, and not to mention a specific guideline to evaluate different commercial Cloud services. Consequently, although we have already learned rich lessons from the evaluation of traditional computing systems [11], [20], it is still necessary to accumulate evaluation experiences in the Cloud Computing domain.
- Evaluation results could be invalid soon after the evaluation and then not reusable. Cloud providers may continually upgrade their hardware and software infrastructures, and new commercial Cloud services may gradually enter the market. Hence, previous evaluation results can be quickly out of date as time goes by. For example, at the time of writing, Google is moving its App Engine service from CPU usage model to instance model [2]; Amazon is still acquiring additional sites for Cloud data center expansion [19]; while IBM just offered a public and commercial Cloud [8]. As a result, customers would have to continually re-design and repeat evaluation for employing commercial Cloud services.
- The back-ends (e.g. configurations of physical infrastructure) of commercial Cloud services are not controllable from the perspective of customers. Unlike consumer-owned computing systems, customers have little knowledge and control over the precise nature of Cloud services even in the "locked down" environment [23]. Evaluations in the context of public Cloud Computing are then inevitably more challenging than that for systems where the customer is in direct control of all aspects [24]. In fact, it is natural that the evaluation of uncontrollable systems would be more complex than that of controllable ones.

Therefore, particularly for commercial Cloud services, it is necessary to find a way to facilitate evaluation, and make existing evaluation efforts reusable and sustainable. This paper suggests an expert system for Cloud evaluation to address the aforementioned issues. This expert system concentrates on processes and experiences rather than results of Cloud services evaluation. When it comes to the general implementation process of Cloud services evaluation, we can roughly draw six common steps following the systematic approach to performance evaluation of computer systems [11], as specified below and illustrated in Figure 1:

(1) First of all, the requirement should be specified to clarify the evaluation purpose, which essentially drives the remaining steps of the evaluation implementation.
(2) Based on the evaluation requirement, we can identify the relevant Cloud service features to be evaluated.
(3) To measure the relevant service features, suitable metrics

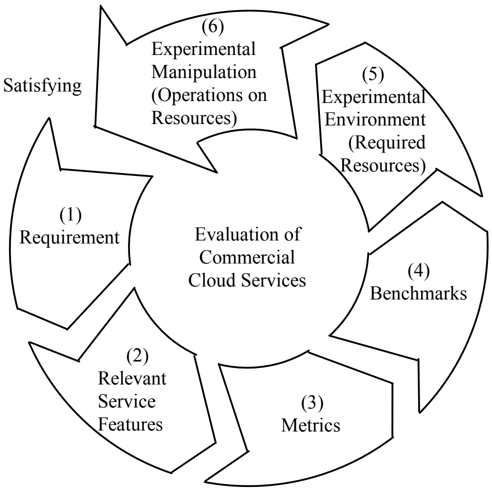

Fig. 1. General process of an evaluation implementation.

should be determined.

(4) According to the determined metrics, we can employ corresponding benchmarks that may already exist or have to be developed.

(5) Before implementing the evaluation experiment, the experimental environment should be constructed. The environment includes not only the Cloud resources to be evaluated but also assistant resources involved in the experiment.

(6) Given all the aforementioned preparation, the evaluation experiment can be done with human intervention, which finally satisfies the evaluation requirement.

Through decomposing and analyzing individual evaluation experiments following the six steps, we have collected and arranged data of detailed evaluation processes. Based on the primary evaluation data, general knowledge about evaluating commercial Cloud services can be abstracted and summarized. After manually constructing the *Data/Knowledge Base*, we can design and implement an *Inference Engine* to realize knowledge and data reasoning respectively. As such, given particular enquiries, the proposed expert system is not only able to supply common evaluation suggestions directly, but also able to introduce similar experimental practices to users for reference.

The remainder of this paper is organized as follows. Section II specifies the establishments of *Data Base*, *Knowledge Base*, and *Inference Engine* in this expert system. Section III employs three samples to show different application cases of this expert system, which also gives our current work a conceptual validation. Conclusions and some future work are discussed in Section IV.

II. Establishment of this Expert System

Similar to general expert systems [10], [12], the expert system proposed in this paper also comprises an *Interface* with which users interact, an *Inference Engine* that performs knowledge/data reasoning, and a *Knowledge Base* that stores

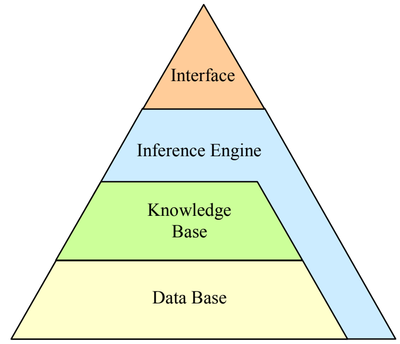

Fig. 2. Structure of this expert system.

common and abstracted knowledge about evaluation of commercial Cloud services. However, we did not employ a specific knowledge acquisition module for building up the *Knowledge Base* in this case. At the current stage, instead of obtaining knowledge by interviewing external experts, we extracted Cloud evaluation knowledge only from the collected data of published experimental studies. Moreover, for the convenience of acquiring experimental references, a *Data Base* is maintained in this expert system to store initially-analyzed details of existing evaluation experiments. The complete structure of this expert system can be illustrated as shown in Figure 2.

Considering the *Interface* of this expert system can be designed at last in the future, this paper only specifies how we are realizing the *Data/Knowledge Base* and *Inference Engine*.

A. Collecting Evaluation Practices (Data Base)

To collect and initially analyze existing evaluation practices, we employed the systematic literature review (SLR) as the main approach. SLR is the methodology applied for Evidence-Based Software Engineering (EBSE) [6], and has been widely accepted as a standard and systematic way to investigate specific research questions by identifying, assessing, and analyzing published primary studies. According to the guidelines of SLR [13], an entire SLR instance mainly requires three stages, namely Preparation, Implementation, and Summarization. After adjusting some steps, here we list a rough SLR procedure suitable for this work:

Planning Review:

- Justify the necessity of carrying out this SLR.
- Identify research questions for this SLR.
- Develop SLR protocol by defining search strategy, selection criteria, quality assessment standard, and data extraction schema for Conducting Review stage.

Conducting Review:

- Exhaustively search relevant primary studies in the literature.
- Select relevant primary studies and assess their qualities for answering research questions.
- Extract useful data from the selected primary studies.
- Arrange and synthesize the initial results of our study into review notes.

Reporting Review:

- Analyze and interpret the initial results together with review notes into interpretation notes.
- Finalize and polish the previous notes into an SLR report.

Due to the limit of space, the detailed SLR process is not elaborated in this paper.[1] With the pre-defined search strategy and rigorous selection criteria, we have identified 46 relevant primary studies covering six commercial Cloud providers from a set of popular digital publication databases (the studies are listed online for reference: http://www.mendeley.com/groups/1104801/slr4cloud/papers/). Note that this work focused only on the commercial Cloud services to make our effort closer to industry's needs. Moreover, this study paid attention to Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) without concerning Software as a Service (SaaS). Since SaaS is not used to further build individual business applications [3], various SaaS implementations may comprise infinite and exclusive functionalities to be evaluated, which could make our SLR out of control even if adopting extremely strict selection/exclusion criteria. After exhaustively identifying evaluation practices, independent evaluation experiments can be first isolated, and then be broken into atomic components by using the data extraction schema. For example, more than 500 evaluation metrics including duplications were finally extracted from the identified Cloud services evaluation studies. The summarized metrics in turn can help facilitate metric selection in future evaluation work [17]. In other words, every single isolated experiment is finally represented as a set of elements, which essentially facilitates knowledge mining among different elements across different evaluation steps. Therefore, the Conducting Review stage in this SLR also indicates the procedure of establishing the *Data Base* of the proposed expert system.

B. Mining Evaluation Experiences (Knowledge Base)

Considering that this expert system gives evaluation suggestions according to users' inputs, the stored knowledge is suitable to be represented as rules each of which is composed of antecedent (to meet input) and consequent (to be output). To efficiently obtain rule-based knowledge from the *Data Base*, we adopted association-rule mining to inspire the method of knowledge mining. By analogy, we can find that the rule-based knowledge used in this expert system has similar characteristics to association rules. As we know, different association rules can express different regularities that underlie a dataset [26]. In other words, association rules have freedom to use and predict any combination of items in a predefined set. As for the expert system, it is supposed that users can start from any particular evaluation step, or from any combination of detailed evaluation steps, to enquire about different evaluation experiences.

When mining association rules in a dataset, we use pre-defined *coverage* to seek combinations of items that appear frequently, and then use *accuracy* to extract suitable rules among each of the identified item combinations [26]. Note that we name an attribute-value pair as an item in the dataset. The *coverage* of a particular item combination refers to the number of data instances comprising the combination, while the *accuracy* of a candidate rule is the proportion of the correctly-predicted data instances to the applied data instances.

Similarly, we followed the same rule-induction procedure to mine evaluation knowledge from the collected experiment data. However, compared with the quantitative and programmable process of rule mining, the knowledge mining in this case has to involve more human interventions. Although some evaluation step details (e.g. the Cloud service features) can be pre-standardized, most of the experiment data (e.g. requirement or experimental manipulation) would be not able to be specified within an exactly same schema from the beginning. As a result, we had to manually extract common knowledge through abstracting specific descriptions of the raw data.

For the convenience of the discussion, we briefly demonstrate the process of knowledge mining only from two-item sets. Suppose the focus is now on the Scalability evaluation of Cloud services. We can initially list all the Scalability-related two-item combinations, gradually abstract their descriptions, and then rationally classify or unify them into fewer groups. At last, common knowledge can be sought within each of the combination groups. Here we only list four straightforward pieces of evaluation experience identified in our work.

- **IF service feature = "*Scalability*" THEN experimental manipulation = "*varying Cloud resource with the same amount of workload*"** (extracted from the change of Cloud resource, without distinguishing that the resource was varied in terms of type or amount).
- **IF service feature = "*Vertical Scalability*" THEN experimental manipulation = "*different types of Cloud resource*"** (extracted from the experiments each of which covers different types of service instances [18], [22]).
- **IF service feature = "*Horizontal Scalability*" THEN experimental manipulation = "*different amount of Cloud resource*"** (extracted by focusing on the amount of the same type of Cloud resource, no matter the resource is CPU core [7] or service instance [16]).
- **IF service feature = "*Scalability*" THEN metric = "*speedup over a baseline*"** (extracted by summarizing Scalability-related metrics like Pipeline Performance Speedup [16], Computation Speedup [22], and Throughput Speedup [18]).

C. Reasoning in Knowledge/Data Base (Inference Engine)

In this expert system, the *Inference Engine* is associated with both *Knowledge Base* and *Data Base*. As such, the *Inference Engine* can not only perform knowledge reasoning but also supply similar experimental cases.

1) First-Order Rule Reasoning in Knowledge Base:

To facilitate the knowledge reasoning process, we proposed to enrich the representations of knowledge by bridging be-

[1]The complete SLR report can be found online: https://docs.google.com/open?id=0B9KzcoAAmi43LV9IaEgtNnVUenVXSy1FWTJKSzRsdw

tween some concepts or item combinations. The bridges can be viewed as common-sense knowledge that supplements the aforementioned, extracted knowledge. For example, we added two new rules **IF service feature = "*Vertical Scalability*" THEN service feature = "*Scalability*"** and **IF service feature = "*Horizontal Scalability*" THEN service feature = "*Scalability*"** to the four previous samples. In fact, the two new rules are always true although they are not generated by knowledge mining. Benefiting from the knowledge bridges, we can employ the algorithm of learning first-order rules [15] to conveniently reveal underlying rule-based knowledge that does not visibly exist in the *Knowledge Base*. In this case, for instance, the expert system can give suggestions of evaluating Vertical Scalability by using a visible rule **IF service feature = "*Vertical Scalability*" THEN experimental environment = "*different types of Cloud resource*"**, and also resorting to the Scalability-related knowledge, as shown below.

- Experimental manipulation = "*varying Cloud resource with the same amount of workload*".
- Experimental environment = "*different types of Cloud resource*".
- Metric = "*speedup over a baseline*".

2) Case Retrieving in Data Base:

As mentioned previously, in addition to evaluation knowledge, this expert system is also able to retrieve experimental cases to users for analogy. In fact, as one of the basic human reasoning processes, analogy is used by almost every individual on a daily basis to solve new problems based upon past experiences. The process of analogy generally follows the procedure of the case-based reasoning (CBR), while one general CBR procedure comprises a four-stage cycle [1], as shown in Figure 3.

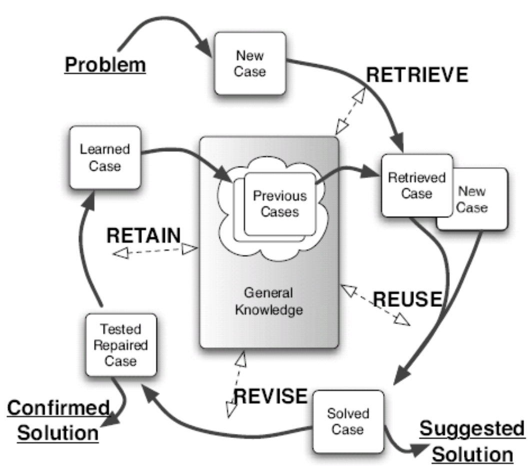

Fig. 3. General cyclical case-based reasoning process.

In the general cyclical CBR process, an initial problem is described as a new case. Following the new case, we can RETRIEVE a case from the previous cases. The retrieved case is combined with the new case through REUSE into a solved case. The REVISE process is then used to test the solution for the new case. Finally, useful experience is RETAINed for future reuse, and the dataset of previous cases will be updated by a new learned case, or by modification of some existing cases.

When it comes to case retrieving, the essential issue is how to identify rational and similar cases to the new one. We proposed three modes of case retrieving in this expert system, namely Precise Mode, Heuristic Mode, and Fuzzy Mode. As the name suggests, under the Precise Mode, the expert system identifies similar evaluation experiments exactly following users' enquiries. For example, suppose a user is interested in the evaluation experiments with respect to Horizontal Scalability of Cloud services, the expert system will only retrieve the experiment data with service feature = "*Horizontal Scalability*"

In the worst case of Precise Mode, there would be no experiment record directly meeting a user's enquiry. The user can then try the Heuristic Mode. The Heuristic Mode relies on the knowledge reasoning process discussed previously. In detail, the expert system first explores the *Knowledge Base* to identify the rules with antecedents meeting the user's enquiry, and then retrieves experiment data that include those rules' consequents. For example, when retrieving data with inquiry service feature = "*Horizontal Scalability*" under the Heuristic Mode, the expert system will list evaluation experiments having experimental environment = "*different amount of Cloud resource*" (according to the previous rule **IF service feature = "*Horizontal Scalability*" THEN experimental environment = "*different amount of Cloud resource*"**). In this case, suppose that even if the retrieved experiments focus on evaluating cost-benefit by using "*different amount of Cloud resource*", they can still be used to inspire the evaluation of Horizontal Scalability.

In the worst case of Heuristic Mode, the expert system could yet retrieve nothing due to lack of data, lack of knowledge, or invalid enquiry. Then, the data retrieving can be switched to the Fuzzy Mode. Ideally, the Fuzzy Mode relies on the uncertain reasoning [12] in the *Knowledge Base*. In the current stage, however, we only realize the Fuzzy Mode to allow the expert system to use sub-content of the enquiry information to explore for useful data. For example, suppose an invalid inquiry includes three elements: service feature = "*Horizontal Scalability*", experimental environment = "*different types of Cloud resource*", and metric = "*speedup over a baseline*". Under the Fuzzy Mode, the expert system first removes one of the inquiry elements, and then uses both Precise Mode and Heuristic Mode methods to identify similar experimental cases. In this sample, consequently, users can still achieve useful experimental cases after removing the inquiry element experimental environment = "*different types of Cloud resource*". Note that, since the case retrieving here is based on incomplete enquiry information, the Fuzzy Mode does not necessarily guarantee that all the retrieved experiments are valuable for users.

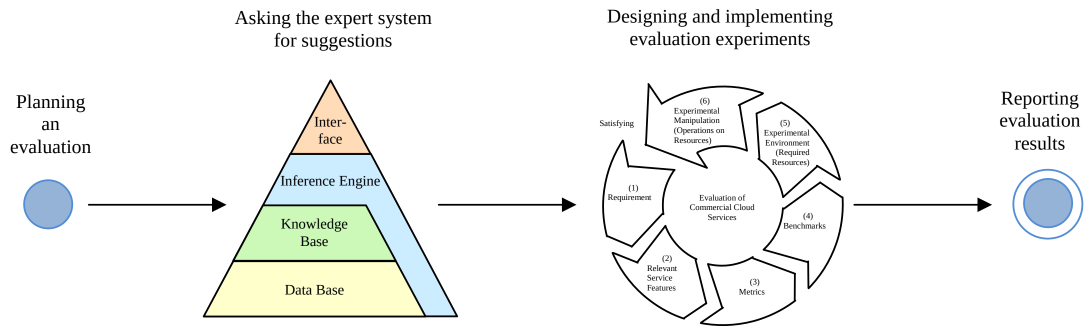

Fig. 4. Position of applying the expert system to Cloud services evaluation.

III. Application of this Expert System

Ideally, this expert system is supposed to deal with enquiries about any component in evaluation experiments. For example, given a particular metric, we can ask the expert system for candidate benchmarks supplying the metric; or given particular experimental operations, we can ask the expert system for what evaluation requirement can be satisfied. Therefore, in general practices of Cloud services evaluation, the proposed expert system can be applied after planning an evaluation and before designing and implementing the evaluation, as illustrated in Figure 4.

In the current stage of our work, we constrain the enquiry condition as: given a particular Cloud service feature, we ask the expert system for suitable evaluation scenarios (the combination of suitable experimental environment and experimental operations) and evaluation metrics (also relevant benchmarks). Here we use three real examples to show the possible application cases of this expert system. The three cases can meanwhile be viewed as a conceptual validation of our current work. Note that, to highlight the application flow, the expert system's working mechanism is simplified without elaborating the data/knowledge reasoning procedures.

A. Application Case 1: Asking for evaluation suggestions before performing analytic modeling.

Analytic modeling is a relatively light-weight evaluation technique, which employs approximate calculations to supply quick and rough analysis [20]. Suppose we decide to adopt analytic modeling to satisfy the requirement about how elastic a particular Cloud platform is [4]. According to the keywords in the requirement, we can find that the concerned Cloud service feature is Elasticity in this case. As such, the requirement is manually translated into "Elasticity" as the input to the expert system. The output includes scenarios and metrics for evaluating Elasticity, as demonstrated in Figure 5. The evaluation scenarios and metrics can be further translated into design parameters and variables [20] by evaluator while performing the modeling work. The suggested metrics like *VM Boosting Latency* are used to model the Cloud platform, while the suggested scenarios like *Workloads rise and fall repeatedly* are used to model the Cloud-hosted workloads. The complete process of this application case is shown in Figure 5.

B. Application Case 2: Asking for evaluation suggestions before performing real measurement.

Real measurement is a relatively effort-intensive evaluation technique, which implements experiments on prototypes or actual computing systems to conduct more accurate analysis [20]. In this case, suppose we plan to measure how variable the real Cloud service performance is [9]. Similarly, the requirement here can be translated into "Variability" as the input to the expert system, and the output suggests scenarios and metrics for evaluating Variability (cf. Figure 6). Unlike the previous case, however, the suggested scenarios like *Repeat experiment at different time* are used to prepare and perform experimental environment/operations, while the suggested metrics like *Standard Deviation with Average Value* are used to measure and display the experimental results. The application flow of this case is shown in Figure 6.

C. Application Case 3: Asking for evaluation suggestions for further decision making about usage strategy of Cloud resources.

This application case is essentially an extension of previous two cases. In fact, suggesting usage strategy of Cloud resources is out of the scope of applying this expert system. This expert system provides suggestions only for Cloud services evaluation, without making any decision based on the evaluation result. Nevertheless, since evaluation is the prerequisite of further decision making, this expert system can still be helpful in this case. Suppose there is an evaluation requirement about choosing "alternative architecture for transaction processing in the Cloud" [14]. This requirement is a typical decision making about alternative strategies of utilizing Cloud resources. Given particular Cloud service features concerned in the predefined architectures, the expert system supplies suggestions of evaluating those service features; the evaluation suggestions can then be employed in each experiment for each of the architectures respectively; the architectures are finally judged through contrasting the evaluation results. In other words, this application case normally comprises a set of sibling experiments with the same evaluation suggestions, and the application flow is similar to Figure 6. As for the sample [14], this expert system will give suggestions of evaluation scenarios and metrics for

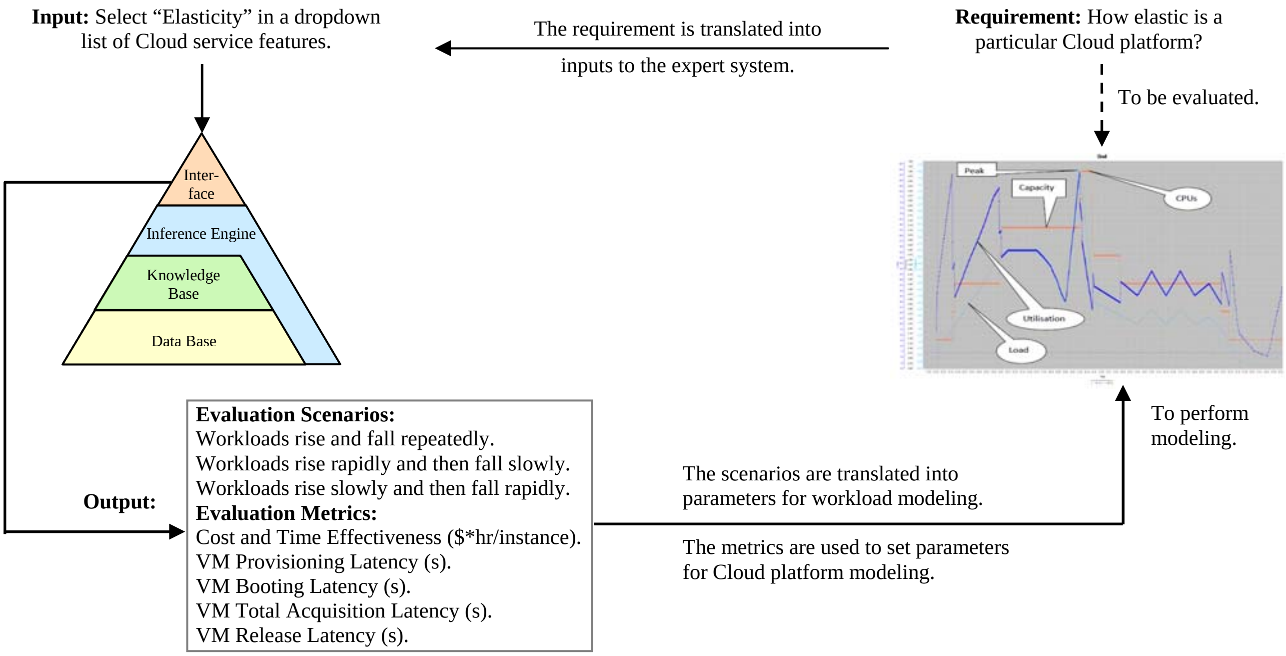

Fig. 5. Asking for evaluation suggestions before performing analytic modeling.

the service features "Storage" and "Cost".

Overall, this section has demonstrated three typical cases of applying the proposed expert system. To better distinguish between these application cases, here we highlight two points:

- ***Direct vs. Indirect Help from the Expert System.*** When it comes to the application context of this expert system, Cases 1 and 2 are in the pure evaluation context, whereas Case 3 is in the decision making context. As previously mentioned, this expert system can facilitate but not directly suggest usage strategies of Cloud resources. Therefore, the expert system directly helps satisfy the evaluation requirement in the first two cases, while it indirectly helps satisfy the evaluation requirement in the third case.
- ***Different Usage Purpose and Sequence of the Evaluation Suggestions.*** In Application Case 1, the suggestion of evaluation metrics is used before the analytic modeling work. In fact, the suggested metrics are used to design the indicator tradeoffs of the simulated Cloud environment. On the contrary, in Application Case 2, the suggestion of evaluation metrics is used to measure the Cloud service indicators and display the result after real experiments.

IV. Conclusions and Future Work

Along with the booming Cloud Computing in industry, various commercial providers have started offering Cloud services. Thus, it is normal and vital to implement evaluations when deciding whether or not to employ a particular Cloud service, or choosing among several candidate Cloud services. However, given the rapidly-changing and customer-uncontrollable conditions, evaluation of commercial Cloud services are inevitably more challenging than that of traditional computing systems. To facilitate evaluation work in the context of Cloud Computing in industry, we proposed to accumulate existing evaluation knowledge, and to establish an expert system for Cloud services evaluation to make evaluation experiences conveniently reusable and sustainable. Note that the proposed expert system does NOT work like an automated evaluation tool or benchmark involved in evaluation implementations, but gives evaluation suggestions or guidelines according to users' enquiries.

The most significant contribution of this work is to help practitioners implement Cloud services evaluation along a systematic way. In fact, it is impossible to require everyone, especially Cloud customers, to be equipped with rich knowledge and expertise on Cloud services evaluation. We can find that the current evaluation work is relatively ad hoc in the Cloud computing field. For example, evaluation techniques and benchmarks are selected randomly. Based on the accumulated evaluation experiences, however, this proposed expert system can intelligently supply rational and comprehensive consultation to future evaluation practices, which has been conceptually validated by using three real application cases.

This paper roughly introduces the structure and components of this expert system, and mainly specifies the study methodology we are following. The methodology then reveals and guides our current and future work, such as using the SLR to collect and analyze existing evaluation practices, following the procedure of association-rule mining to extract evaluation knowledge, building the *Inference Engine* to conduct knowledge reasoning and data retrieving, and patching a well-designed *Interface* to complete the expert system.

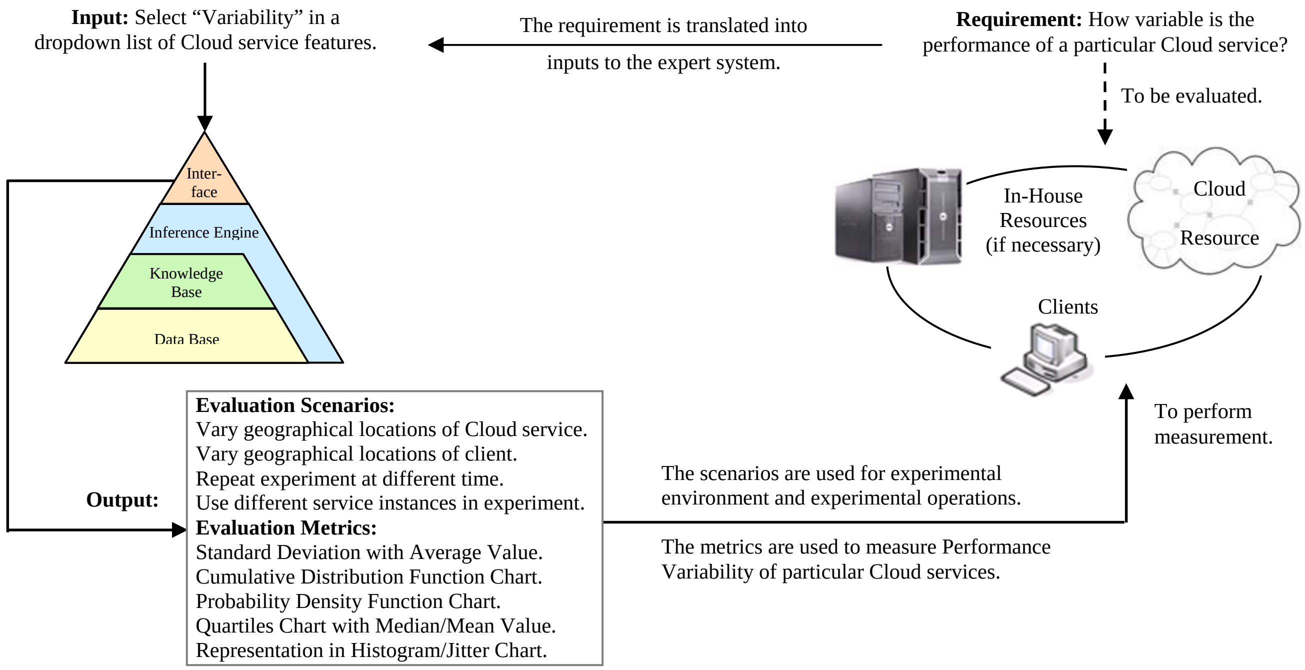

Fig. 6. Asking for evaluation suggestions before performing real measurement.

The prototypes of different function parts will be gradually developed and integrated into an online system.[2] Finally, a data maintenance system will be also built up online to collect feedback, update data, and keep all the data versions with time stamps. The *Data/Knowledge Base* of the expert system can eventually be updated regularly through the maintenance system.

Acknowledgment

This project is supported by the Commonwealth of Australia under the Australia-China Science and Research Fund.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, Mar. 1994, pp. 39–59.

[2] G. D. Alesandre, "Updated app engine pricing FAQ!," *Google Groups*, available at https://groups.google.com/forum/#!msg/google-appengine/Hluog1_a3n4/uFMhaBWhVi8J, Jun. 2011.

[3] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow? Towards a benchmark for the Cloud," *Proc. 2nd Int. Workshop on Testing Database Systems (DBTest 2009) in conjunction with ACM SIGMOD/PODPS Int. Conf. Management of Data (SIGMOD/PODS 2009)*, ACM Press, Jun. 2009, pp. 1–6.

[4] P. Brebner, "Is your Cloud elastic enough?," *MeasureIT*, vol. 82, no. 7, 2011, pp. 1–7.

[2]The current version can be viewed as a knowledge lookup system: http://sites.google.com/site/expertsystemforcloudevaluation/

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comp. Sy.*, vol. 25, no. 6, Jun. 2009, pp. 599–616.

[6] T. Dybå, B. A. Kitchenham, and M. Jørgensen, "Evidence-based software engineering for practitioners," *IEEE Softw.*, vol. 22, no. 1, Jan. 2005, pp. 58–65.

[7] C. Evangelinos and C. N. Hill, C.N. "Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2," *Proc. 1st Workshop on Cloud Computing and Its Applications (CCA '08)*, Oct. 2008, pp. 1–6.

[8] D. Harris, "Watch out, world: IBM finally offers a real Cloud," *GigaOM - Tech News, Analysis and Trends*, available at http://gigaom.com/cloud/watch-out-world-ibm-finally-offers-a-real-cloud/, Apr. 2011.

[9] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production Cloud services," Delft Univ. Technol., Netherlands, Tech. Rep. PDS-2010-002, 2010.

[10] P. Jackson, *Introduction to Expert Systems*, 3rd ed. England, UK: Addison Wesley Longman Limited, 1999.

[11] R. K. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, NY: Wiley Computer Publishing, John Wiley & Sons, Inc., Apr. 1991.

[12] S. L. Kendal and M. Creen, *An Introdection to Knowledge Engineering*. London, UK: Springer-Verlag, 2007.

[13] B. A. Kitchenham and S. Charters, "Guidelines for performance systematic literature reviews in software engineering," Keele Univ. and Univ. Durham, UK, EBSE Tech. Rep. EBSE-2007-01, ver. 2.3, 2007.

[14] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the Cloud," *Proc. 2010 ACM SIGMOD Int. Conf. Management of Data (SIGMOD '10)*, ACM Press, Jun. 2010, pp. 579–590.

[15] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, Nov. 2006, pp. 159–190.

[16] J. Li, M. Humphrey, D. Agarwal, K. Jackson, C. van Ingen, and

Y. Ryu, "eScience in the Cloud: A MODIS satellite data reprojection and reduction pipeline in the Windows Azure platform," *Proc. 23rd IEEE Int. Symp. Parallel and Distributed Processing (IPDPS 2010)*, IEEE Computer Society, Apr. 2010, pp. 1–10.

[17] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics for evaluating commercial Cloud services," *Proc. 13th ACM/IEEE Int. Conf. Grid Computing (Grid 2012)*, Sept. 2012, to appear.

[18] W. Lu, J. Jackson, and R. Barga, "AzureBlast: A case study of developing science applications on the Cloud," *Proc. 1st Workshop on Scientific Cloud Computing (Science Cloud 2010)*, ACM Press, Jun. 2010, pp. 413–420.

[19] R. Miller, "A look inside Amazon's data centers," *Data Center Knowledge*, available at http://www.datacenterknowledge.com/archives/2011/06/09/a-look-inside-amazons-data-centers/, Jun. 2011.

[20] M. S. Obaidat and N. A. Boudriga, *Fundamentals of Performance Evaluation of Computer and Telecommunication Systems*.Hoboken, New Jersey: John Wiley & Sons, Inc., Jan. 2010.

[21] R. Prodan and S. Ostermann, "A survey and taxonomy of infrastructure as a service and Web hosting Cloud providers," *Proc. 10th IEEE/ACM Int. Conf. Grid Computing (GRID 2009)*, IEEE Computer Society, Oct. 2009, pp. 17–25.

[22] J. J. Rehr, F. D. Vila, J. P. Gardner, L. Svec, and M. Prange, "Scientific computing in the Cloud," *Comput. Sci. Eng.*, vol. 12, no. 3, May/Jun. 2010, pp. 34–43.

[23] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, and D. Patterson, "Cloudstone: Multi-platform, multi-language benchmark and measurement tools for Web 2.0," *Proc. 1st Workshop on Cloud Computing and Its Applications (CCA '08)*, Oct. 2008, pp. 1–6.

[24] V. Stantchev, "Performance evaluation of Cloud computing offerings," *Proc. 3rd Int. Conf. Advanced Engineering Computing and Applications in Science (ADVCOMP '09)*, IEEE Computer Society, Oct. 2009, pp. 187–192.

[25] J. Stokes, "The PC is order, the Cloud is chaos," *Wired Cloudline, Wired.com*, available at http://www.wired.com/cloudline/2011/12/the-pc-is-order/, Dec. 2011.

[26] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, Jun. 2005.

[27] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, May 2010, pp. 7–18.